

Simultaneous Embedding of Planar Graphs with Few Bends

Cesim Erten and Stephen G. Kobourov

Department of Computer Science
University of Arizona
{cesim,kobourov}@cs.arizona.edu

Abstract

We consider several variations of the simultaneous embedding problem for planar graphs. We begin with a simple proof that not all pairs of planar graphs have simultaneous geometric embeddings. However, using bends, pairs of planar graphs can be simultaneously embedded on the $O(n^2) \times O(n^2)$ grid, with at most three bends per edge, where n is the number of vertices. The $O(n)$ time algorithm guarantees that two corresponding vertices in the graphs are mapped to the same location in the final drawing and that both the drawings are without crossings.

The special case when both input graphs are trees has several applications, such as contour tree simplification and evolutionary biology. We show that if both input graphs are trees, only one bend per edge is required. The $O(n)$ time algorithm guarantees that both drawings are crossings-free, corresponding tree vertices are mapped to the same locations, and all vertices (and bends) are on the $O(n^2) \times O(n^2)$ grid ($O(n^3) \times O(n^3)$ grid).

For the special case when one of the graphs is a tree and the other is a path we can find simultaneous embeddings with fixed-edges. That is, we can guarantee that corresponding vertices are mapped to the same locations and that corresponding edges are drawn the same way. We describe an $O(n)$ time algorithm for simultaneous embeddings with fixed-edges for tree-path pairs with at most one bend per tree-edge and no bends along path edges, such that all vertices (and bends) are on the $O(n) \times O(n^2)$ grid, ($O(n^2) \times O(n^3)$ grid).

Article Type	Communicated by	Submitted	Revised
Regular paper	E. R. Gansner and J. Pach	December 2004	August 2005

1 Introduction

Traditional problems in graph drawing involve the layout of a single graph, whereas in simultaneous graph drawing we are concerned with the layout of multiple related graphs. Embedding planar graphs simultaneously is motivated by problems in graph thickness and geometric thickness, and applications such as contour tree simplification and visualization of graphs that evolve through time. In addition to generalizing the notion of planarity, techniques for simultaneous embedding of cycles have been used to show that degree-4 graphs have geometric thickness at most two [12].

In many different settings it is useful to visualize related graphs, that is, graphs that are defined on the same set of vertices. Software engineering, databases, and social network analysis, are all examples of areas where multiple relationships on the same set of objects are often studied. In evolutionary biology, phylogenetic trees are used to visualize the ancestral relationship among groups of species. Depending on the assumptions made, different algorithms produce different phylogenetic trees. Klingner and Amenta [18] and Munzner *et al* [20] present techniques for visualization of such trees. Comparing the outputs and determining the most likely evolutionary hypothesis can be difficult if the drawings of the trees are laid out independently of each other.

Contour trees were proposed by van Kreveld *et al* [26] for computing isolines on terrain maps in geographic information systems. Carr, Snoeyink and van de Panne [6] use contour trees for scientific and medical visualization. Contour tree simplification applies the ideas of topological persistence to trees and is another application for simultaneous drawing of trees [5]. Simultaneous embedding techniques are also useful in the visualization of graphs that evolve through time, for example, in the context of visualization of the evolution of software [9].

Consider the case where a pair of related graphs is given and the goal is to compare the graphs by visualizing them. When examining a graph the user constructs a mental view of it, for example, using the positions of the vertices relative to each other. If drawings for the two graphs are obtained independently, there would be little correspondence between the two layouts, since the viewer has no *mental map* between the two graphs. When viewing multiple graphs the user has to reconstruct the mental map after examining each graph and our goal should be to aid the user in this reconstruction while providing a readable drawing for each graph individually.

In simultaneous graph embedding, the vertices are placed in the exact same locations in all the graphs. Fixing the vertex positions in all the graphs preserves the mental map, but at the expense of readability of the individual drawings, if edges are to be drawn with straight-line segments. With this in mind, in this paper we consider the problem of drawing planar graphs on the same point-set using few bends. We describe efficient algorithms for simultaneous drawing of pairs of general planar graphs on small integer grids. We also describe better results for pairs of trees and tree-path pairs.

1.1 Previous Work

Given a single planar graphs, the existence of drawings with straight-line segments and no crossings is well known [14, 24, 27]. Tutte [25] extended these results to shown that every 3-connected planar graph has a convex drawing. These techniques, however, do not guarantee anything about the resolution of the drawing and thus are not well-suited for automated graph drawing. The vertex resolution problem was addressed by de Fraysseix, Pach and Pollack [10] and Schnyder [23] who showed that any n -vertex planar graph can be drawn with straight-line segments and no crossings using $O(n^2)$ area, with vertices placed at integer grid points.

The problem of *simultaneous geometric embedding* of two or more graphs is more recent. It is known that a simultaneous geometric embedding of an n vertex 3-connected planar graph and its dual can be found in $O(n)$ time using $O(n^2)$ area [13]. Brass *et al.* [3] describe linear time algorithms for simultaneous geometric embeddings of pairs of paths, cycles, and caterpillars, also using $O(n^2)$ area; see Fig. 1. Cappos and Kobourov [4] show how to simultaneously embed tree-path pairs, such that the tree is drawn without crossings, using one straight-line segment per edge, and the path is drawn without crossings, using one circular arc segment per edge.

A related problem is the problem of *graph thickness*, defined as the minimum number of planar subgraphs into which the edges of the graph can be partitioned; see the survey by Mutzel *et al.* [21]. If a graph has thickness two then it can be drawn on two layers such that each layer is without crossings and the corresponding vertices of different layers are placed in the same locations. Dillencourt, Eppstein and Hirschberg [11] study the *geometric thickness* of graphs, where the edges are required to be straight-line segments. Thus, if two graphs have a simultaneous geometric embedding, then their union has geometric thickness two. Similarly, the union of any two planar graphs has graph thickness two. Duncan *et al.* [12] use simultaneous geometric embedding techniques to show that degree-four graphs have geometric thickness two.

While the thickness and simultaneous embedding problems are related, results from one do not necessarily translate into the other. Bose, Hurtado, Rivera-Campo and Wood [2] show that the complete convex graph K_{2n} can be partitioned into n plane spanning trees and moreover, characterize all the different partitions. In particular, they show that K_{2n} can be partitioned into n non-crossing paths. However, given n paths, it is not always possible to embed them simultaneously for $n \geq 3$, as shown by Brass *et al.* [3].

Simultaneous drawing of multiple graphs is also related to the problem of *fixed point-set embedding* of planar graphs. Bose [1] and Gritzman *et al.* [15] show that if the mapping between the vertices V and the points P is not fixed, then trees and outer-planar graphs can be drawn without crossings, using straight-line edges. In the same setting general planar graphs cannot be drawn without bends. If bends are allowed, Kaufmann and Wiese [17] show that two bends per edge suffice. However, if the mapping between V and P is predetermined, Pach and Wenger [22] show that $O(n)$ bends per edge are sufficient to

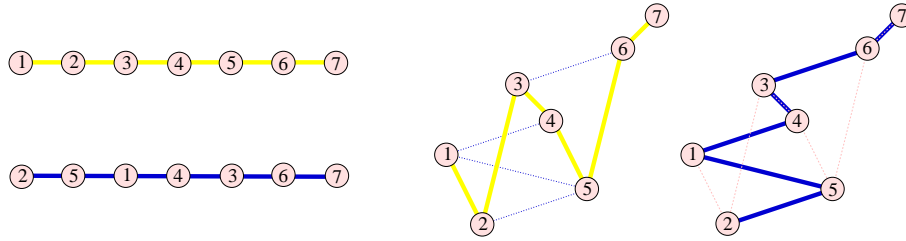


Figure 1: Two paths simultaneously embedded such that one path is x -monotone and the other is y -monotone.

guarantee planarity, where n is the number of vertices in the graph. They also show that this bound on the number bends per edge is tight.

1.2 Our Contributions

Formally, the drawing \mathcal{D} of a graph $G = (V, E)$ is a function that maps each vertex $u \in V$ to a distinct point $\mathcal{D}(u)$ in the plane, and each edge $(u, v) \in E$ to a simple Jordan curve $\mathcal{D}(u, v)$ with endpoints $\mathcal{D}(u)$ and $\mathcal{D}(v)$. The problem of simultaneously embedding two planar graphs G_1, G_2 is the problem of finding drawings D_1, D_2 with corresponding vertices of G_1 and G_2 mapped to the same points in the plane, such that each drawing has no crossings.

In this paper we study the following three variations of the simultaneous embedding problem, depending on the way the edges of the graphs are drawn:

Definition 1 Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ the *simultaneous geometric embedding of G_1 and G_2* is the problem of finding plane straight-line drawings D_1 and D_2 of G_1 and G_2 , respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2 .

Definition 2 Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ the *simultaneous embedding of G_1 and G_2 with consistent edges* is the problem of finding plane drawings D_1 and D_2 of G_1 and G_2 , respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2 and every shared edge $e \in E_1 \cap E_2$ is represented with the same simple open Jordan curve in D_1 and D_2 .

Definition 3 Given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ the *simultaneous embedding of G_1 and G_2* is the problem of finding plane drawings D_1 and D_2 of G_1 and G_2 , respectively, such that every vertex is mapped to the same point in the plane in both D_1 and D_2 .

The definitions are inclusive in the given order: simultaneous geometric embedding is a special case of simultaneous embedding with consistent edges, which is in turn a special case of simultaneous embedding.

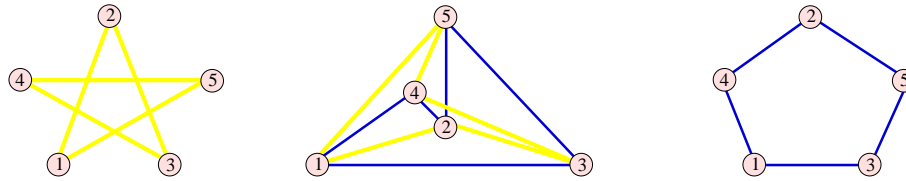


Figure 2: The union of the graph on the left and the graph on the right is K_5 , but the middle drawing shows a simultaneous geometric embedding of the two graphs

In Section 2 we begin with a simple proof that not all pairs of planar graphs have a simultaneous geometric embedding. Next, we present an $O(n)$ time algorithm for the simultaneous embedding of pairs of planar graphs on the $O(n^2) \times O(n^2)$ grid, with at most three bends per edge, where n is the number of vertices. If bend-vertices are placed on grid points then the $O(n^3) \times O(n^3)$ grid suffices.

In Section 3 we show that if both the input graphs are trees, only one bend per edge is required. The linear time and area bounds still apply. We also describe an $O(n)$ time algorithm for simultaneous embeddings with consistent edges for tree-path pairs. The algorithm places the vertices (and the bend-points) on the $O(n) \times O(n^2)$ grid ($O(n^2) \times O(n^3)$ grid) and there is at most one bend per tree-edge and no bends along the path edges.

In Section 4 we briefly discuss the implementation of these algorithms, show some of the resulting layouts, and conclude with several open problems.

2 Simultaneous Embeddings

Simultaneous geometric embeddings are easy to find on small integer grids for pairs of simple graphs such as paths, cycles, and caterpillars [3]. A common method for simultaneous embedding of such graph pairs is to determine an ordering for the vertices in each graph, and then place the vertices such that the locations of the vertices appearing in the determined ordering are increasing monotonically in some direction in the plane. This method is illustrated for the case when both graphs are paths in Fig. 1.

For pairs of general planar graphs, and even for pairs of outer-planar graphs, simultaneous geometric embeddings do not always exist. This is the main motivation for relaxing the conditions of simultaneous geometric embeddings, to just simultaneous embeddings, by dropping the straight-line edge constraint. Under these weaker constraints, we can obtain simultaneous drawings with few bends per edge. Such drawings are also useful for pairs of trees, as it is not known whether the simultaneous geometric embedding of pairs of trees is always possible.

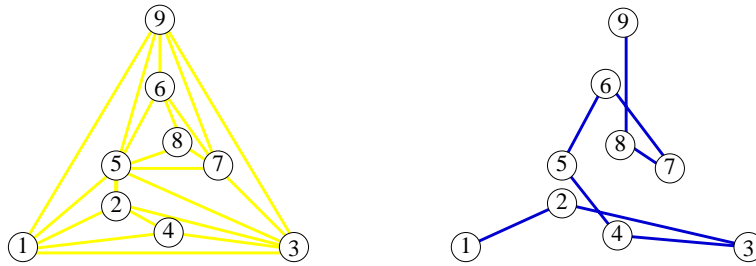


Figure 3: A planar graph G and a path P that do not allow a simultaneous geometric embedding.

2.1 Simultaneous Geometric Embeddings

While it may be tempting to say that if the union of two graphs contains a subdivision of K_5 or $K_{3,3}$ then the two graphs have no simultaneous geometric embedding, this is not the case; see Fig. 2. In fact, while planarity testing for a single graph can be done in linear time [16], the complexity of determining whether a pair of planar graphs admits a simultaneous geometric embedding is not known.

However, it is known that there exist pairs of planar graphs that cannot be simultaneously embedded [3]. Here we briefly describe a simple case of a pair of planar graphs that do not admit a simultaneous geometric embedding.

Theorem 1 *There exist a planar graph G and a path P , such that there is no simultaneous geometric embedding of G and P .*

Proof: Consider graph G and path P as shown in Fig. 3. Let G' be the subgraph of G induced on vertices $\{1, 2, 3, 4, 5\}$, and G'' be the subgraph of G induced on vertices $\{5, 6, 7, 8, 9\}$. Since G is 3-connected fixing the outer face fixes an embedding for G . With the given outer face of G , the path P contains two crossings: one involving $(4, 5)$, and the other one involving $(8, 9)$. Graph G' has six faces and unless we change the outer face of G' such that it contains the edge $(1, 2)$ or $(2, 3)$, the edge $(4, 5)$ is involved in a crossing in the path. Similarly for G'' , unless we change its outer face such that it contains $(5, 6)$ or $(6, 7)$, the edge $(8, 9)$ is involved in a crossing in the path. However G' and G'' do not share any faces and removing both crossings depends on taking two different outer faces, which is impossible. Thus, regardless of the choice for the outer face of G , path P contains a crossing. \square

2.2 Relaxing the Constraints

While some classes of planar graphs allow simultaneous geometric embeddings, there are other classes that do not, and still others for which it is not known

whether simultaneous geometric embeddings exist. Since the latter two categories contain a large number of planar graph classes (trees, outer-planar graphs, general planar graphs), it is natural to look for simultaneous drawings with weaker constraints. One possible solution for larger classes of graphs is to relax the constraints on the edges. Instead of restricting the edges to be straight-line segments we allow each edge to be drawn as a sequence of straight-line segments. Recall that such embeddings are called *simultaneous embeddings* (rather than simultaneous geometric embeddings).

Note that it is always possible to find a simultaneous embedding of any two planar graphs, if we are willing to accept a large number of bends per edge. Given a point-set P of size n in the plane and a planar graph G with n vertices, together with a one-to-one mapping between the vertices of G and the points in P , we can find drawings of G on P using edges with bends and no crossings [22]. This allows us to embed any number of planar graphs simultaneously. However, the resulting drawings contain $O(n)$ bends per edge. Next, we describe methods to simultaneously embed any two planar graphs so that each edge has at most three bends.

2.3 Simultaneous Embedding with Few Bends

Since in this version of the problem we no longer insist on straight-line edges, the problem of simultaneously embedding two graphs boils down to finding a point-set in the plane and a mapping between the vertices and the points, with as few bends per edge as possible. The following theorem summarizes our results for pairs of general planar graphs.

Theorem 2 *Given two planar graphs G_1 and G_2 on the same vertex set, we can simultaneously embed G_1 and G_2 using at most three bends per edge. The resulting drawing requires an integer grid of size $O(n^2) \times O(n^2)$ such that each vertex is placed on a grid point, and the algorithm requires $O(n)$ time, where n is the number of vertices.*

Proof: Initially, we assume the input graphs are 4-connected. We show how to remove this assumption using the technique of Kaufmann and Wiese [17] later in the proof.

Vertex Placement: Assuming that the graphs G_1 and G_2 are 4-connected, we place the vertices in a monotonically increasing x and y order, similar to that of Brass *et al* [3].

First we find a Hamiltonian cycle H_1 of G_1 and a Hamiltonian cycle H_2 of G_2 . We can do this in linear time using the algorithm of Chiba and Nishizeki [8]. Starting at an arbitrary vertex in H_1 we traverse its vertices, assigning increasing x -coordinates to each vertex visited. Starting at a random vertex in H_2 we traverse its vertices, assigning increasing y -coordinates to each vertex visited. Not considering the final edges enclosing the cycles, this gives us an x -monotone path for H_1 and a y -monotone path for H_2 ; see Fig. 4(a).

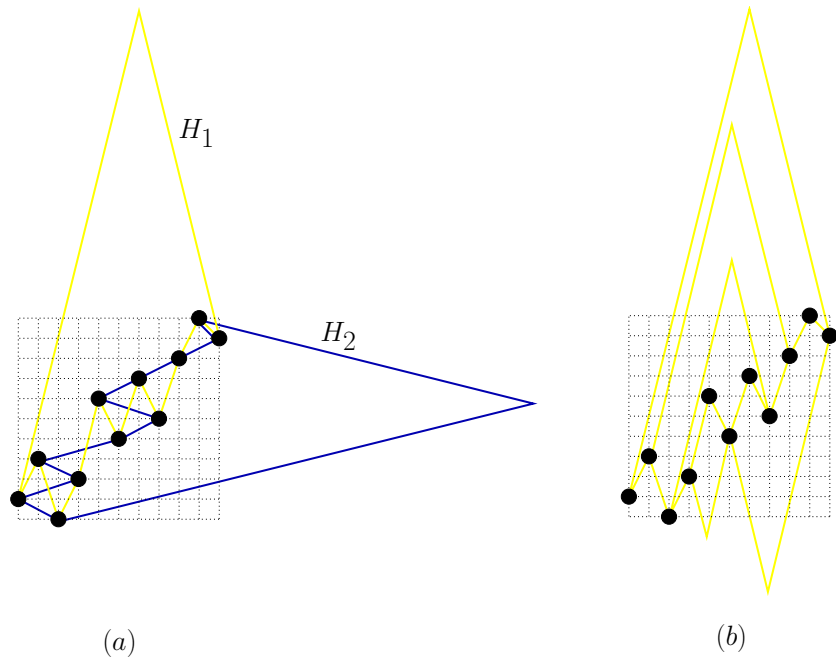


Figure 4: (a) H_1 and H_2 drawn simultaneously. (b) Only the edges of G_1 are shown. The edges inside the Hamiltonian cycle H_1 have the same slope as the outermost edge and go inside the cycle. Similarly the edges outside the cycle in the embedding are drawn outside the cycle.

Since both paths are monotone, the edges of each paths do not intersect. Let δ be the largest slope of the edges on the path defined by H_1 . We complete the drawing of the cycle H_1 by drawing the final edge between the leftmost vertex and the rightmost vertex. It is drawn with two segments such that the slope of the initial segment starting at the leftmost vertex is δ' and the slope of the second segment ending at the rightmost vertex is $-\delta'$, where δ' is slightly larger than δ . Since G_1 is Hamiltonian, the cycle H_1 divides the edges into two groups: inside and outside edges (with respect to H_1). Then each of the inside edges is drawn with two line segments with slopes δ' and $-\delta'$ on the inside of H_1 . Similarly, the outside edges are drawn with the same slopes on the outside of H_1 ; see Fig. 4(b).

The edges of G_2 are handled in the same way with respect to H_2 . It is easy to see that the vertex set requires grid size $n \times n$. The overall area of the drawing is larger, as the bend points lie outside the original grid. Since $\delta' \leq n$, the horizontal and vertical distance between a bend point and the two endpoints is at most n^2 . Thus the entire drawing fits inside an $O(n^2) \times O(n^2)$ grid.

Making the Graphs 4-connected: For the case when the input graphs are not 4-connected, we use the techniques introduced in [17] to augment them. Given

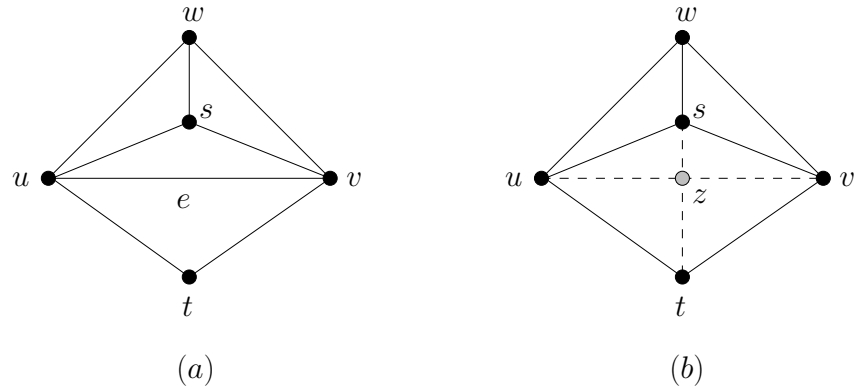


Figure 5: Removing separating triangles: (a) Edge e is part of the separating triangle (u, v, w) . The two faces containing e are (u, v, s) and (u, v, t) . (b) The separating triangle is removed by deleting e , introducing z and connecting it to u, v, s , and t .

a planar graph we first fully triangulate it by adding extra edges if necessary. Next we make the graph 4-connected by introducing new vertices. This is done by removing all the separating triangles in G , where a separating triangle is a cycle of length 3 such that the removal of the vertices of the cycle disconnects G . Separating triangles of G can be easily found by another algorithm of Chiba and Nishizeki [7]. Let $e = (u, v)$ be an edge of a separating triangle in G such that e is adjacent to the faces (u, v, s) and (u, v, t) ; see Fig. 5. We remove the separating triangle by inserting a dummy vertex z on e , deleting the edge e , and introducing four new edges: $(u, z), (v, z), (s, z), (t, z)$. The newly introduced vertex z is not part of any separating triangle, so each time we introduce such a vertex we decrease the number of separating triangles. Doing the same operation on all the separating triangles gives us a 4-connected planar graph.

Once G_1 and G_2 have been augmented to 4-connected graphs, we obtain the Hamiltonian cycles H_1 and H_2 of G_1 and G_2 . We augment the edges of H_2 with the extra vertices of G_1 , and augment the edges of H_1 with the extra vertices of G_2 . The placement of the Hamiltonian cycles and the drawing of the remaining edges is done as before. After finishing the placement, we treat the dummy vertices as bend points and ignore the edges inserted in the augmentation phase. As a result, an edge $e = (u, v)$ that got split with a dummy vertex z ends up having at most three bend points: one between u and z , one at the location of z , and finally one between v and z . As there are $O(n)$ dummy vertices, the bounds for the integer grid remain unchanged.

Running Time: The two non-trivial operations are finding the separating triangles and finding the Hamiltonian cycles. Finding the separating triangles and making the graphs 4-connected takes linear time [7]. A Hamiltonian cycle in a 4-connected planar graph can also be found in linear time [8]. \square

The corollary below follows from the above theorem by fixing the slopes of all the edges and refining the grid.

Corollary 3 *Given two planar graphs G_1 and G_2 on the same vertex set, we can simultaneously embed G_1 and G_2 using at most three bends per edge on an integer grid of size $O(n^3) \times O(n^3)$, with all the vertices and bend-points at grid-points.*

Proof: Consider the original $n \times n$ grid where H_1 and H_2 are placed. Let the slope $\delta = n$, where δ and $-\delta$ are the slopes of all edge segments among edges drawn with bends. Let $e = (u, v) \in G_1$ such that u is placed to the left of v and e is drawn with a bend point p . Let x_{dist}, y_{dist} be the x -coordinate and y -coordinate distances between u and v . The x -coordinate distance between u and the point p is $(n \times x_{dist} - y_{dist})/2n$. If we place a $2n \times 2n$ grid inside each unit square of the original grid, then the x -coordinate distance between u and p is an integer. Since the slope of the segment \overline{up} is n , the y -coordinate distance between u and p is also an integer, and p is on a grid point. Similar argument applies to the edges of G_2 as well. The final grid area is $O(n^3) \times O(n^3)$. \square

3 Simultaneous Embeddings for Trees and Paths

The special case when the input graphs are trees has several applications, such as contour tree simplification and evolutionary trees. The generic algorithm from the previous section can be modified to simultaneously embed tree-tree pairs and tree-path using fewer bends. For the case of tree-tree pairs we show that only one bend per edge is needed. For the case when one of the trees is a path, we can do even better, by guaranteeing that path edges have no bends at all.

3.1 Tree-Tree Pairs

The theorem below follows from Theorem 2 and the above corollary.

Theorem 4 *Given two trees T_1 and T_2 on the same vertex set, they can be simultaneously embedded in linear time, using at most one bend per edge, on an integer grid of size $O(n^2) \times O(n^2)$ (or $O(n^3) \times O(n^3)$, if both the vertices and bend-points are on grid points).*

Proof: We first show that we can augment the two trees to Hamiltonian planar graphs with Hamiltonian cycles H_1, H_2 , respectively. Since we do not introduce any new vertices in the augmentation stage, the number of bends required will be at most one per tree edge. The placement of the vertices is the same as that in Theorem 2 and Corollary 3 to obtain the $O(n^2) \times O(n^2)$ and $O(n^3) \times O(n^3)$, respectively.

Given a tree T we show how to construct H_T using a recursive divide-and-conquer procedure: the input to the recursive call is a subtree T and the output

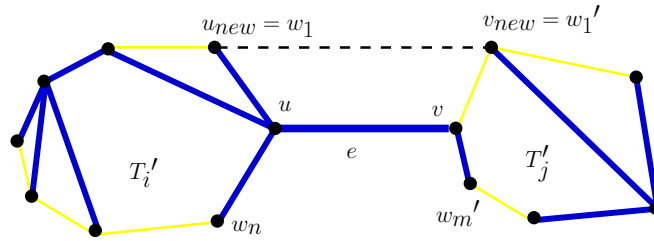


Figure 6: Constructing the Hamiltonian cycle H_T from H_i and H_j . Dark edges are tree edges and the others are augmentation edges. We add the tree edge $e = (u, v)$ and the edge (u_{new}, v_{new}) (for the initial case where T_i and T_j are singletons the augmentation edge is parallel to e) to obtain T' .

is the Hamiltonian cycle H_T and the modified graph T' . We augment the tree T with edges until the resulting graph T' has a Hamiltonian cycle H_T . The base case for the recursion is a tree with just one node, $T = \{u\}$. In this case, let $H_T = (u, u)$, and $T' = T$. For all other cases, we take an arbitrary edge $e = (u, v)$ from T . Let T_i, T_j be the two trees obtained after the removal of e from T . Assume we can construct Hamiltonian cycles H_i and H_j of T_i and T_j , respectively. Let T_i' and T_j' be the graphs that we get after these constructions, corresponding to T_i and T_j , respectively. We merge the two subgraphs into the new graph $T' = T_i' \cup T_j'$ by adding e to T' .

In order to combine the Hamiltonian cycles of the two subgraphs into a Hamiltonian cycle for union, we need to add one more edge between the two subgraphs. We add an edge between a neighbor u_{new} of u to a neighbor v_{new} of v and combine the two cycles by dropping the edges (u, u_{new}) and (v, v_{new}) if they exist. Note that if we have two parallel edges to be dropped we drop only one of them.

Let $H_i = (u, w_1, w_2, \dots, w_n, u)$ and $H_j = (v, w_1', w_2', \dots, w_m', v)$. If T_i' has only one vertex u we assign $u_{new} = u$, and if it has two vertices u and u' we assign $u_{new} = u'$. We do similar assignments for v_{new} if T_j' has one or two vertices. In order to find u_{new}, v_{new} for all other cases, we can pick w_1 or w_n as u_{new} and w_1' or w_m' as v_{new} . Without loss of generality, let $u_{new} = w_1$ and $v_{new} = w_1'$. Then the new Hamiltonian cycle becomes, $H_T = (u, v, w_m', w_{m-1}', \dots, w_1', w_1, w_2, \dots, w_n, u)$; see Fig. 6.

Planarity: The above recursive procedure augments the tree T to a graph T' that has a Hamiltonian cycle. We still need to show that the resulting graph T' is planar. Recall the recursive procedure above and let us assume that T_i' and T_j' are planar. Then there exists a planar embedding for T_i' so that the edge (u, w_1) is on the outer face and a planar embedding for T_j' so that the edge (v, w_1') is on the outer face. Since all the vertices u, w_1, v, w_1' are on the outer faces of their graphs, the inserted edges (u, v) and (w_1, w_1') do not have any crossings with the edges of T_i' and T_j' . The resulting graph T' is planar, and the resulting embedding is a planar embedding.

Running Time: We only need to show that the Hamiltonian cycle construction takes linear time, since the rest of the algorithm is the same as the one described in Theorem 2. Note that we do not have to explicitly find planar embeddings of T_i' and T_j' at each level of the recursion. The planar embedding of the final graph T' suffices. We can find the planar embedding of T' in linear time [16]. The merging of the two Hamiltonian cycles requires constant number of operations at each recursive step and thus the overall running time of the algorithm is $O(n)$. \square

3.2 Tree-Path Pairs with Consistent Edges

Note that although the algorithm for tree-tree pairs simultaneously embeds two trees with the corresponding vertices mapped on the same positions (preserves the mental map for the vertex set), there is a significant drawback in terms of the mental map of the edges. In particular, edges common to both graphs are drawn differently in the two drawings unless they happen to be on the paths defined by the Hamiltonian cycles. Simultaneous embeddings with consistent edges requires that shared edges be represented the same way in both drawings. We describe an algorithm for simultaneous embedding with consistent edges for a tree and a path below.

Theorem 5 *Given a tree T and a path P on the same vertex set, they can be simultaneously embedded with consistent edges in linear time, using at most one bend per edge, on an integer grid of size $O(n) \times O(n^2)$ (or $O(n^2) \times O(n^3)$, if both the vertices and bend-points are on the grid).*

Proof: The main idea is the same as that in Theorem 4, except that we ensure that the edges common to both T and P are in the Hamiltonian cycle for the tree. Then the path and the Hamiltonian cycle (minus an edge) have a simultaneous geometric embedding. The rest of the tree edges are routed as before, thus yielding a simultaneous embedding with consistent edges for T and P .

Let $E_{T,P}$ be the set of edges common to both T and P . In order to obtain a Hamiltonian cycle for the tree T similar to the construction in Theorem 4, we augment it with edges until the resulting graph T' has a Hamiltonian cycle $H_{T'}$. This time we make sure that $H_{T'}$ contains all edges that are in common with the path. Again the base case for the recursion is a tree with just one node, $T = \{u\}$. In this case, let $H_T = (u, u)$, and $T' = T$. For all other cases, we take an edge $e = (u, v) \in E_{T,P}$ from T if such an edge exists. If not, we take an arbitrary edge $e = (u, v) \in T$. Let T_i, T_j be the two trees obtained after the removal of e from T . Assume we can construct Hamiltonian cycles H_i and H_j of T_i and T_j , respectively. Let T_i' and T_j' be the graphs that we get after these constructions, corresponding to T_i and T_j , respectively. We merge the two subgraphs into the new graph $T' = T_i' \cup T_j'$ by adding e to T' .

In order to combine the Hamiltonian cycles of the two subgraphs into a Hamiltonian cycle for union, we need to add one more edge between the two

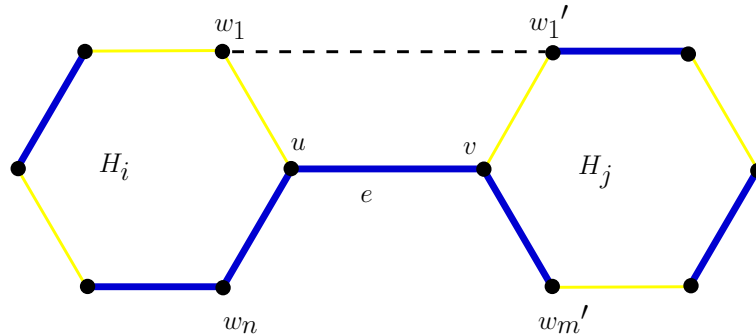


Figure 7: Constructing the Hamiltonian cycle H_T from H_i and H_j . The dark edges belong to the path, while the others are tree edges or augmentation edges. Vertices u and v can be incident to only one path edge in H_i and H_j , respectively. Without loss of generality, let $(u, w_n) \in E_{T,P}$ and let $(v, w_m') \in E_{T,P}$. Then we insert the edge $e = (u, v)$ and the edge (w_1, w_1') to obtain T' .

subgraphs. We add an edge between a neighbor u_{new} of u to a neighbor v_{new} of v and combine the two cycles by dropping the edges (u, u_{new}) and (v, v_{new}) if they exist.

Let $H_i = (u, w_1, w_2, \dots, w_n, u)$ and $H_j = (v, w_1', w_2', \dots, w_m', v)$. If T_i' has only one vertex u we assign $u_{new} = u$, and if it has two vertices u and u' we assign $u_{new} = u'$. We do similar assignments for v_{new} if T_j' has one or two vertices. In order to find u_{new}, v_{new} for all other cases, we check the first and the last edges of the Hamiltonian cycles.

Assume $e = (u, v) \notin E_{T,P}$. Then no edge in T_i, T_j is in $E_{T,P}$, since an edge $e \notin E_{T,P}$ is picked only if there is no edge $e \in E_{T,P}$ in the current subgraph. In this case we assign u_{new} to either w_1 or w_n arbitrarily. Similar arbitrary assignment is done for v_{new} .

Now assume $e = (u, v) \in E_{T,P}$. Since P is a path, either $(u, w_1) \notin E_{T,P}$, or $(u, w_n) \notin E_{T,P}$ (otherwise, vertex u must have degree greater than 2 in the path). Without loss of generality, assume $(u, w_1) \notin E_{T,P}$. We assign $u_{new} = w_1$. The same holds for H_j , that is, either $(v, w_1') \notin E_{T,P}$ or $(v, w_m') \notin E_{T,P}$. Without loss of generality, assume $(v, w_1') \notin E_{T,P}$. We assign $v_{new} = w_1'$. As a result of this insertion the new Hamiltonian cycle becomes, $H_T = (u, v, w_m', w_{m-1}', \dots, w_1', w_1, w_2, \dots, w_n, u)$; see Fig. 7.

Planarity of the new graph T' and the running time of the algorithm follow from the arguments in the proof of Theorem 4. □

4 Conclusion and Future Work

We implemented the algorithms described above using the LEDA library [19] in C++. Fig. 8 and Fig. 9 show some of the resulting layouts for a path and tree and

two trees, respectively. Note that when several edges not on the Hamiltonian cycle leave a vertex, they may all overlap; see edges (1,3), (1,4), and (1,6) on Fig. 8. This problem can be addressed by perturbing the bend-points as in [17].

All of the algorithms in this paper rely on the approach of augmenting planar graphs to Hamiltonian planar graphs, so as to obtain simultaneous embeddings. However, for a simultaneous embedding with consistent edges, this technique cannot be extended from the tree-path pair case to pairs of trees (and hence cannot be extended to larger classes of planar graphs). The reason is that two arbitrary trees cannot always be augmented to graphs with Hamiltonian cycles that contain all the common edges. Thus, if we would like to represent the common vertices and edges the same way for both graphs, different techniques are needed.

Several problems regarding simultaneous embeddings remain open:

- Given a pair of trees, is it always possible to find a simultaneous geometric embedding?
- Given a tree-path pair, is it always possible to find a simultaneous geometric embedding?
- Given a pair of trees, is it always possible to find a simultaneous embedding with consistent edges such that the number of bends is a small constant?
- Given a pair of planar graphs, what is the complexity of determining whether they admit a simultaneous geometric embedding?

Acknowledgments

We would like to thank Michael Kaufmann and David Eppstein for the comments about the problems discussed in the paper, the anonymous referees for the ideas about improving the area bounds, and Petr Moravsky for helping with the implementation.

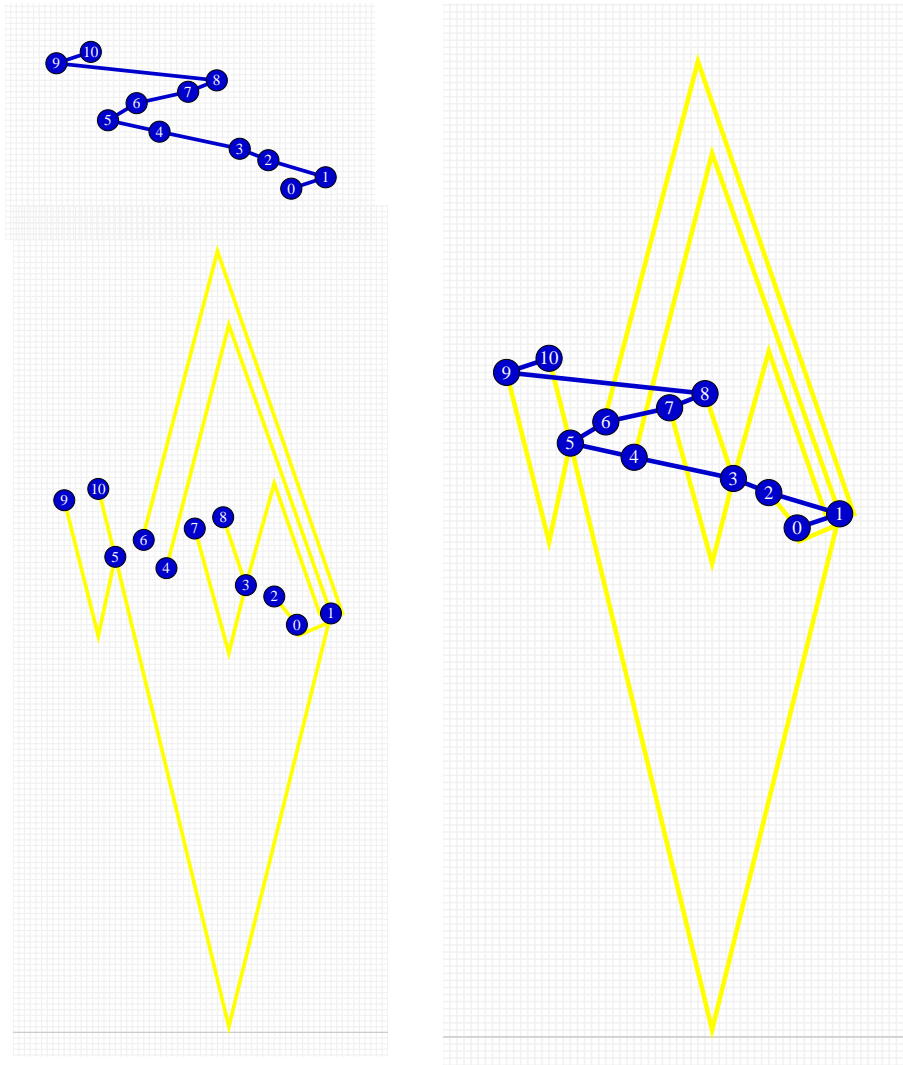


Figure 8: A simultaneous embedding with consistent edges for a tree and a path. The path $(0, 1, \dots, 10)$ is shown on the top left. The tree is shown on the bottom left. Note that the path and the tree share the edge $(0,1)$. The combined view of the tree and the path is shown on the right.

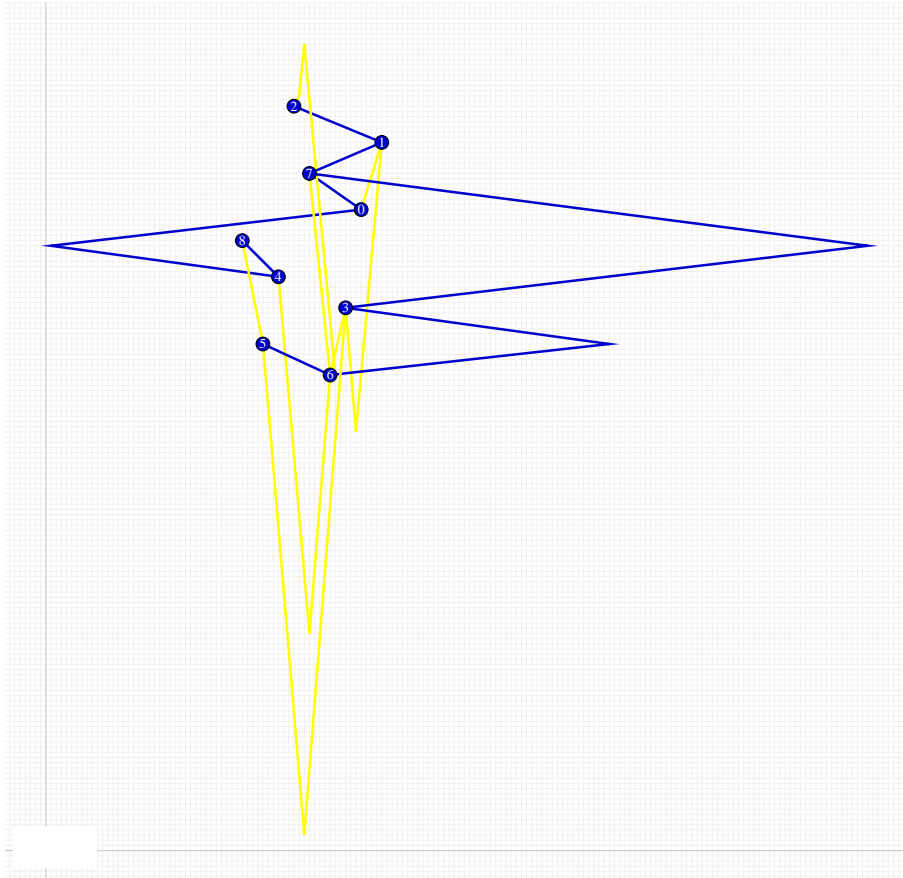


Figure 9: A simultaneous embedding for two trees. The tree T_1 (in blue) consists of the edges: $(5,6)$, $(3,6)$, $(3,7)$, $(7,0)$, $(7,1)$, $(1,2)$, $(0,4)$, $(4,8)$ and the tree T_2 (in yellow) consists of the edges: $(8,5)$, $(5,3)$, $(3,1)$, $(3,6)$, $(1,0)$, $(6,4)$, $(6,7)$, $(6,2)$.

References

- [1] P. Bose. On embedding an outer-planar graph in a point set. *Computational Geometry: Theory and Applications (CGTA)*, 23(3):303–312, 2002.
- [2] P. Bose, F. Hurtado, E. Rivera-Campo, and D. Wood. Partitions of complete geometric graphs into plane trees. In *12th International Symposium on Graph Drawing (GD '04)*, pages 71–81, 2004.
- [3] P. Brass, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous graph embedding. *Computational Geometry: Theory and Applications (CGTA)*. To appear in 2005.
- [4] J. Cappos and S. G. Kobourov. Trees on tracks. Technical Report TR04-14, University of Arizona, Department of Computer Science, 2004.
- [5] H. Carr, J. Snoeyink, and M. van de Panne. Contour tree simplification with local geometric measures. In *14th Annual Fall Workshop on Computational Geometry*, pages 51–52, 2004.
- [6] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization*, pages 497–504, 2004.
- [7] N. Chiba and T. Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14:210–223, 1985.
- [8] N. Chiba and T. Nishizeki. The hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *Journal of Algorithms*, 10(2):187–211, 1989.
- [9] C. Collberg, S. G. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *ACM Symposium on Software Visualization (SoftVis)*, pages 77–86, 2003.
- [10] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.
- [11] M. B. Dillencourt, D. Eppstein, and D. S. Hirschberg. Geometric thickness of complete graphs. *Journal of Graph Algorithms and Applications*, 4(3):5–17, 2000.
- [12] C. A. Duncan, D. Eppstein, and S. G. Kobourov. The geometric thickness of low degree graphs. In *20th Annual ACM-SIAM Symposium on Computational Geometry (SCG)*, pages 340–346, 2004.
- [13] C. Erten and S. G. Kobourov. Simultaneous embedding of a planar graph and its dual on the grid. In *13th Intl. Symp. on Algorithms and Computation (ISAAC)*, pages 575–587, 2002.

- [14] I. Fáry. On straight lines representation of planar graphs. *Acta Scientiarum Mathematicarum*, 11:229–233, 1948.
- [15] P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified points. *American Math. Monthly*, 98:165–166, 1991.
- [16] J. Hopcroft and R. E. Tarjan. Efficient planarity testing. *Journal of the ACM*, 21(4):549–568, 1974.
- [17] M. Kaufmann and R. Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1):115–129, 2002.
- [18] J. Klingner and N. Amenta. Case study: Visualization of evolutionary trees. In *IEEE Symposium on Information Visualization (INFOVIS)*, pages 71–74, 2002.
- [19] K. Mehlhorn and S. Näher. *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge University Press, 1999.
- [20] T. Munzner, F. Guimbretiere, S. Tasiran, L. Zhang, and Y. Zhou. Tree-juxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Transactions on Graphics*, 22(3):453–462, 2003.
- [21] P. Mutzel, T. Odenthal, and M. Scharbrodt. The thickness of graphs: a survey. *Graphs and Combinatorics*, 14(1):59–73, 1998.
- [22] J. Pach and R. Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17:717–728, 2001.
- [23] W. Schnyder. Planar graphs and poset dimension. *Order*, 5(4):323–343, 1989.
- [24] S. K. Stein. Convex maps. *Proceedings of the American Mathematical Society*, 2(3):464–466, 1951.
- [25] W. T. Tutte. How to draw a graph. *Proc. London Math. Society*, 13(52):743–768, 1963.
- [26] M. J. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *13th Annual ACM-SIAM Symposium on Computational Geometry (SCG)*, pages 212–220, 1997.
- [27] K. Wagner. Bemerkungen zum vierfarbenproblem. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 46:26–32, 1936.